

RECEIVED
CENTRAL FAX CENTERIN THE CLAIMS

MAY 08 2007

Please amend claims 1, 3-7, 9-13 and 15-18 as follows:

1. (CURRENTLY AMENDED) A method for providing contextual diagnostic data at a point of failure of a software program, comprising:
 - (a) registering callbacks for one or more modules and sub-applications within the program;
 - (b) examining a call stack for the program upon failure of the program;
 - (c) notifying the registered callbacks for the modules and sub-applications based on the examined call stack;
 - (d) performing callback processing, wherein the notified callbacks of the modules and sub-applications extract and supply the contextual diagnostic data; [[and]]
 - (e) packaging the contextual diagnostic data supplied by the notified callbacks of the modules and sub-applications; and
 - (f) using the packaged contextual diagnostic data for further analysis in order to troubleshoot the point of failure of the software program.
2. (ORIGINAL) The method of claim 1, wherein the registering step (a) comprises registering callbacks for the modules and sub-applications when an address of a procedure or function within the modules and sub-applications is on the call stack upon the failure of the program.
3. (CURRENTLY AMENDED) The method of claim 1, wherein the contextual diagnostic data is comprised of stack data, heap data, global data or external data.
4. (CURRENTLY AMENDED) The method of claim 1, ~~wherein the packaging step (e)~~ comprises further comprising storing the packaged contextual diagnostic data.
5. (CURRENTLY AMENDED) The method of claim 1, ~~wherein the packaging step (e)~~ comprises further comprising transferring the packaged context data to a server computer.

6. (CURRENTLY AMENDED) The method of claim 5, ~~wherein the packaging step (e) comprises further comprising~~ storing the transferred, packaged contextual diagnostic data on the server computer.

7. (CURRENTLY AMENDED) An apparatus for providing contextual diagnostic data at a point of failure of a software program, comprising:

a computer; and

logic, performed by the computer, for:

(a) registering callbacks for one or more modules and sub-applications within the program;

(b) examining a call stack for the program upon failure of the program;

(c) notifying the registered callbacks for the modules and sub-applications based on the examined call stack;

(d) performing callback processing, wherein the notified callbacks of the modules and sub-applications extract and supply the contextual diagnostic data; [[and]]

(e) packaging the contextual diagnostic data supplied by the notified callbacks of the modules and sub-applications; and

(f) using the packaged contextual diagnostic data for further analysis in order to troubleshoot the point of failure of the software program.

8. (ORIGINAL) The apparatus of claim 7, wherein the logic for registering (a) comprises logic for registering callbacks for the modules and sub-applications when an address of a procedure or function within the modules and sub-applications is on the call stack upon the failure of the program.

9. (CURRENTLY AMENDED) The apparatus of claim 7, wherein the contextual diagnostic data is comprised of stack data, heap data, global data or external data.

10. (ORIGINAL) The apparatus of claim 7, ~~wherein the logic for packaging (e) comprises further comprising~~ logic for storing the packaged contextual diagnostic data.

11. (CURRENTLY AMENDED) The apparatus of claim 7, ~~wherein the logic for packaging~~ ~~(e) comprises~~ further comprising logic for transferring the packaged contextual diagnostic data to a server computer.

12. (CURRENTLY AMENDED) The apparatus of claim 11, ~~wherein the logic for packaging~~ ~~(e) comprises~~ further comprising logic for storing the transferred, packaged contextual diagnostic data on the server computer.

13. (CURRENTLY AMENDED) An article of manufacture comprising a program storage device embodying [[logic]] instructions that, when read and executed by a computer, cause the computer to perform a method for providing contextual diagnostic data at a point of failure of a software program, ~~the logic comprising:~~

- (a) registering callbacks for one or more modules and sub-applications within the program;
- (b) examining a call stack for the program upon failure of the program;
- (c) notifying the registered callbacks for the modules and sub-applications based on the examined call stack;
- (d) performing callback processing, wherein the notified callbacks of the modules and sub-applications extract and supply the contextual diagnostic data; [[and]]
- (e) packaging the contextual diagnostic data supplied by the notified callbacks of the modules and sub-applications; and
- (f) using the packaged contextual diagnostic data for further analysis in order to troubleshoot the point of failure of the software program.

14. (ORIGINAL) The article of claim 13, wherein the registering step (a) comprises registering callbacks for the modules and sub-applications when an address of a procedure or function within the modules and sub-applications is on the call stack upon the failure of the program.

15. (CURRENTLY AMENDED) The article of claim 13, wherein the contextual diagnostic data is comprised of stack data, heap data, global data or external data.

16. (CURRENTLY AMENDED) The article of claim 13, ~~wherein the packaging step (e)~~
~~comprises~~ further comprising storing the packaged contextual diagnostic data.

17. (CURRENTLY AMENDED) The article of claim 13, ~~wherein the packaging step (e)~~
~~comprises~~ further comprising transferring the packaged contextual diagnostic data to a server
computer.

18. (CURRENTLY AMENDED) The article of claim 17, ~~wherein the packaging step (e)~~
~~comprises~~ further comprising storing the transferred, packaged contextual diagnostic data on the
server computer.